THE COGNITIVE COMPUTER

ON LANGUAGE, LEARNING, AND ARTIFICIAL INTELLIGENCE

Roger C. Schank

with

Peter G. Childers

THE COGNITIVE COMPUTER

ON LANGUAGE, LEARNING, AND ARTIFICIAL INTELLIGENCE

Roger C. Schank
with
Peter G. Childers

ADDISON-WESLEY PUBLISHING COMPANY, INC.
READING, MASSACHUSETTS MENLO PARK, CALIFORNIA
WOKINGHAM, BERKSHIRE AMSTERDAM DON MILLS, ONTARIO SYDNEY

Library of Congress Cataloging in Publication Data

Schank, Roger C., 1946–
The cognitive computer.

Includes index.
1. Artificial intelligence. 2. Electronic digital computers—Programming. I. Childers, Peter G. II. Title.
Q335.S386 1984 001.64 84-12326
ISBN 0-201-06443-X

Cover design by Marshall Henrichs
Set in 11 point Palatino by Waldman Graphics, Inc., Pennsauken, NJ

ABCDEFGHIJ-DO-8654
First printing, October 1984

PREFACE

"When *I* use a word," Humpty Dumpty said, in a rather scornful tone, "it means just what I choose it to mean—neither more nor less."

"The question is," said Alice, "whether you *can* make words mean so many different things."

"The question is," said Humpty Dumpty, "which is to be master—that's all."

—*Through the Looking Glass*

A book on computers may seem a strange place to find a discussion of the nature of human thinking and language, but Artificial Intelligence is a strange field. This book is about the nature of human intelligence and what it would mean to have machine intelligence.

In this computer age, questions about what computers can do ought to be considered in parallel with questions about what

people can do. There are three important questions for anyone who now uses or who will someday use a computer:

1. What do we have to know about computers in order to live in a world that is full of them?
2. What can we learn about what it means to be intelligent through our development of computers that can understand?
3. How will intelligent computers affect the world we live in?

This book is my attempt to address these three questions. I try to set the computer in perspective, to see it as a machine, a machine with fantastic possibilities. I attempt to allay the fears that those not knowledgeable about computers have upon entering the computer age. To put the computer into the right perspective, we must look at humans. When we think that computers understand, we must consider this idea in light of what we imagine to be the case when humans understand. Knowing how to program in BASIC may not be as important a skill in the computer age as knowing just how advanced intelligent machines are likely to get. In this book I try to give enough information on the problems facing people who are trying to make machines intelligent, so that people who will be affected by these machines will know what is likely to come when.

I am interviewed by various newspapers, magazines, or television shows on the average of once a week. When I decided to get a Ph.D., the last thing on my mind was a career as a media star. I wanted to get far away from the concerns of the average man. I did not want to have to work on something that bored me. I wanted to have fun in what I did, to be challenged, and, most of all, to be in that ivory tower I had read about. I wanted to be a professor because a life centered on ideas appealed to me. It seemed reasonable that a professor at a university would have the opportunity to create ideas without regard for their ultimate utility. Because I was fascinated by the workings of the mind, I decided to explore how people think. I did not enter psychology because the experimental nature of that discipline did not appeal to me. I did not want to run tightly controlled experiments on how people remembered lists of nonsense syllables. I wanted to

know how people communicated; how they created new ideas; how they understood the ideas of others. As time went on I found myself, after traveling a rather crooked path, in the field of Artificial Intelligence (AI). What this field encompassed was not agreed upon exactly by any two workers in AI. I was drawn to it because a great many very bright people were in it; because it dealt with computers, which were both new (and therefore exciting) and logical (and thus free of the fuzziness that characterizes so much nonscientific thinking), and because it was legitimate within this field to speculate upon the nature of the mind.

Now, almost 20 years later, I am a professor in a major university, the head of an important AI laboratory, and I do get to work on and think about all the ideas I had hoped I could. My life, however, is not solely concerned with ideas. Two major differences exist between the life I had imagined and the life I live. First, I am not unconcerned with business and money. In fact, I am the president of a private company in the business of selling AI programs. Second, I am not working in an obscure field having to do with the mind. What I say and do is printed quite regularly in the media.

This book, then, has two purposes. First, its intent is to inform the public about the subject of Artificial Intelligence, not from the perspective of a science-oriented journalist, who may or may not understand what he or she has seen and read, but from the viewpoint of one who is involved deeply in the subject. Second, it seems important to ponder the reasons why my obscure field has hit the front pages. The public has discovered AI but is not quite sure what it is. The fact that AI has become a public issue is both important and somewhat sad. It says more about our society than one would realize at first glance.

Despite the massive publicity that Artificial Intelligence has received, the general public does not have a very good idea of what the subject is all about. We have a vague sense of intelligent computers that eventually will take over the world; we even have a general fear or excitement about the possibilities of this happening. Between the opinion that such an occurrence is impossible, and the opinion that it already has happened or is just about to happen, there is a large gap. What is the reality? Is HAL from 2001 already here, or is it reasonable to expect that he will never

get here? Is AI just a subject for pointy-headed academics, or will every person need some understanding of AI and computers in general just to be able to cope?

These are the issues that now are discussed by newspapers and by people at cocktail parties. On the one hand, it is nice that people care about such things. On the other, it seems a bit odd that so many people want to discuss their opinions about a subject on which they have so little concrete information.

So, in this book I will try to do what most academics abhor. I will try to boil down twenty years of work on a technical subject into a few pages presented in a nontechnical way. In doing so I hope to inform the public about what the reality of AI is and where the pitfalls and possibilities lie. To do this, I have oversimplified complex ideas, not given credit where credit is due, and generally committed a whole set of what, to an academic, are unpardonable sins. I apologize to my colleagues in AI beforehand, and hope they agree that the goal of informing the public makes bending some of the rules worthwhile.

I do want to mention, however, the numerous people who have worked with me and supported me over the years. Included herein are a set of programs and ideas that took years to develop. As I could not have done very much of what I did without their input I would like to thank these individuals here.

In my early years, I derived much knowledge and counsel from Jacob Mey, Sheldon Klein, and Eugene Pendergraft. At Stanford, Ken Colby and Jerry Feldman were very important to my intellectual development. The ideas and programming ability of my research team at Stanford were critical in changing my own view of the world. The members of that team were Chris Riesbeck, Chuck Rieger, Neil Goldman, Sylvia Russell, Linda Hemphill, and numerous others. At Yale, my students were the heart of my research team. Specifically, I want to mention my Yale Ph.D.s, Jim Meehan, Wendy Lehnert, Rich Cullingford, Bob Wilensky, Jaime Carbonell, Anatole Gershman, Gerry De Jong, Mallory Selfridge, Rick Granger, Mike Lebowitz, and Janet Kolodner. Other people who were not my students, but who were critical in developing ideas and programs, were Steve Shwartz and Mike Dyer. Also, many of my current students have been significant in their help, specifically, Larry Birnbaum, Natalie Dehn, Margot Flowers, Gregg Collins, and many others. A special word

of thanks is given to my Yale colleagues who have helped over the years, specifically Bob Abelson, who developed the script concept, and a great deal more, with me.

My work has required a significant amount of financial support, which has come, for the most part, from the U.S. government, specifically the Department of Defense. When one mentions the Department of Defense in a research funding context, there is the usual groan of how scientists are helping make more weapons and such. In fact, the Department of Defense, through the Advanced Research Projects Agency, the Office of Naval Research, and more recently, the Air Force, have been some of the most enlightened supporters of real scientific research in this country. I thank the people who have made those agencies as sensible and significant as they are. I also thank the National Science Foundation for its support over the years.

Various people have taken great pains to help me say what I want to say when I write. Earlier drafts of this book were read and commented on by Larry Birnbaum, Gregg Collins, Kris Hammond, Natalie Dehn, Chris Riesbeck, Diane Schank, Steve Slade, Jim Meehan, and Margot Flowers. Three people spent an extra amount of time helping me. I especially thank Bill Purves, Larry Hunter, and Ann Drinan for their aid.

Last of all, I would like to mention that Yale University is a magnificent place to be a part of. They have put up with me in all my various incarnations, and although it is hard to thank an institution, I hereby do so.

New Haven, Connecticut
March 1984

CONTENTS

PART

ONE

CHAPTER ONE

DON'T START THE COMPUTER REVOLUTION WITHOUT ME

Suddenly, computers are everywhere: in the schools, in the banks, in the offices, in the automobiles—even in toys. And just as suddenly, everyone is feeling "computer illiterate." Advertisements for personal computers imply that if we don't rush to buy a computer for our children they will fail in school and never land a job. Adults who have lived happily for years without knowing anything about computers are starting to feel as if they have been left in the stone age. The media bombard the average person with the idea that if he doesn't learn about computers, he is going to be left behind. We are told, even if only implicitly, that computers are frightening devices bound to control our lives unless we learn how to program them. On top of this, the media has begun to sensationalize the imminent arrival of intelligent machines, and apparently has decided that they are going to be *Japanese* intelligent machines, to boot.

This book has one basic message: stop worrying. The computer revolution has not passed you by. In fact, quite the opposite is the case. The computer revolution hasn't caught up to where you are. If you can't use today's computers without pain, then just wait. It is the computers that will have to change, not you.

One might expect a computer scientist to urge people to learn a programming language as soon as possible and to recommend that they purchase the best and most powerful home computer they can afford. A typical personal computer book reminds the reader that one must become "computer literate" in order to succeed, and suggests that the best way to do this is to buy the book and learn whatever computer skills it happens to teach. These books urge you to learn about computers today in order to prepare for the world of tomorrow. But if tomorrow's computers are even worth thinking about, much less waiting for, it is because they will *not* require their users to be "computer literate."

Consider cars and televisions, two machines that play an important role in our daily lives. The main reason they are so useful is that we can use them without having to know anything about the technology that produced them. Computers are likely to affect us in the same ways that TVs and cars have: They will provide entertainment and services, and we will wonder how we ever lived without them. The average person will not have to learn how to program computers any more than he has to learn auto mechanics or TV repair. It's nice to be able to fix your own TV and to rebuild your car's engine; but you wouldn't worry if someone called you *television-illiterate* or *automobile-illiterate* because you happened to be part of the vast majority of people who *don't* fix their own cars and TVs. *Computers intimidate us because we imagine they are smarter than we are.* But a computer really can't do anything unless you, with your human intelligence, tell it exactly what you want it to do, down to the last detail. Anyone who thinks today's computers are intelligent in the full sense of the word has no appreciation of the extraordinary power of the human mind, much less of the real possibility of developing programs that approach human levels of intelligence. If the computers of the future gain any semblance of intelligence it will be because we have begun to unravel some of the mysteries of human intelligence and model them on the computer.

COMPUTER LITERACY

What should we know about computers? One thing seems clear: As computers get better at reflecting human thought processes, they also should become as easy to use as televisions—we will

just turn them on and type to them as we would to another person. It is not necessary, therefore, to worry too much about today's computers. Early computing skills will go the way of early car-repair skills, a relic of a period when the users of a new machine had to be experts or devoted hobbyists. Computer literacy will seem as absurd a term as television literacy.

What is "computer literacy," anyway, and what distinguishes someone who is "computer literate" from someone who isn't? Computer literacy means different things to different people. For some it means being able to drop the words *bit* and *byte* into a conversation about technology. To others it means being able to use a word processor. To people who are serious about it, it means knowing programming languages such as BASIC, Pascal, LISP, or APL, and knowing something about computer structure and function. People who know these languages can do only one thing that people who don't know them cannot do: They can write programs. But what kind of programs can they write? Most of today's programming languages are oriented toward mathematical and statistical operations. Computers have been programmed for these kinds of operations since they were first developed, mainly because engineers and mathematicians were the original developers of computers. But the average person doesn't have mathematical and statistical operations to perform, so even if everyone learned these programming languages, they would have no reason to write programs in them. The average person might become *computer literate,* but he would have nothing to say.

An in-depth knowledge of computers, programming languages, and programming techniques is almost irrelevant for the average person. Unless you have some programs that you really need to write, you needn't bother learning a programming language.

There is one very good reason to learn programming, but it has nothing to do with preparing for high-tech careers or with making sure one is computer literate in order to avoid being cynically manipulated by the computers of the future. The real value of learning to program can only be understood if we look at learning to program as an exercise of the intellect, as a kind of modern-day Latin that we learn to sharpen our minds.

The centerpiece of any computer program is the algorithm it uses to accomplish its task. An algorithm is quite simply a

recipe, a set of step-by-step instructions for completing a process of some kind. (The problem of processes and algorithms is examined in Chapter 4.) Learning how to formulate an efficient algorithm in order to solve a problem sharpens the mind. Programming can develop your intellect and show you the benefits (and problems) of step-by-step reasoning. If you want to learn programming as an intellectual adventure, fine; the process certainly will develop your reasoning skills.

The media can't seem to get away from the idea that we are about to make a sudden leap from the Apple computer to some version of the HAL 9000 electronic monster depicted in Arthur C. Clarke's *2001: A Space Odyssey*. In order to understand what leap we are talking about here we should take another, less apocalyptic, look at the computer and the average person.

THE COMPUTER AND YOU

The main reason for the average person to care about computers is that they might contribute to daily life by providing some kind of service. We all know, for example, that computers allow large businesses and institutions of every kind to keep their records straight, with incredible speed and efficiency. Sadly enough, the average person only becomes aware that some business is using a computer to handle transactions when something goes wrong. Computers can break down, just as any of our machines can. Their circuits can overheat and deteriorate, and they can be destroyed by magnetic fields and fluctuations in electrical power. There's the story of the broken electronic banking machine that ate the same customer's bank card and refused to let him make withdrawals or deposits three times in a row. The customer finally pulled out a revolver and shot the machine six times. However, billing errors that arrive in the mail are rarely computer errors; they are usually errors on the part of some human involved in programming or data entry. Somewhere along the line, the wrong instructions were given—a decimal place was moved or a key was pressed twice by accident.

These things don't happen to everybody; nor do they happen all the time. Most of us never have to be bothered with the

knowledge that all our purchases, banking, insurance, medical records, air travel, telephone service, and other utilities are kept track of, and properly handled by, computers. Computers contribute to our ease and well-being, since we don't have to stand in line or wait eight weeks for a phone to be installed and connected, and can be billed easily and promptly for exactly what we owe.

A COMPUTER IN EVERY HOME: SIDE-EFFECTS

Some of the best and most appropriate uses of computers in the home are really side-effects of computer technology: games, word processing, and sending personal letters over computer networks. Computer scientists have had such toys available to them for years. What has changed is that now computers are small and cheap. The average person can have his very own computer sitting on a desk at home. He can take advantage of the side-effects of the computer age that computer scientists have developed for their own use.

What else can someone who isn't a programmer do with a computer? Why isn't he using the real power of computers, namely, to run programs that perform a necessary task? Why are games, word processing, and computer mail the primary things that today's computers can do? Does the lack of *computer literacy* prevent the average person from doing more with the computer? What can a computer-literate person do with a home computer?

Today's computers can only be addressed in severely restricted and prescribed ways. This is why someone who wants a computer to make everyday life easier in some way will quicky become frustrated. *Computers do only what they have been programmed to do, and this essential fact of computers will not change.* Any intelligence computers may have will result from an evolution of our ideas about the nature of intelligence—not as a result of advances in electronics.

To understand this distinction we must discuss the difference between *hardware* and *software*. A good analogy here comes from television. When you buy a television, it is only as inter-